Stats(data, default, use_copy, is_sorted)

The ``Stats`` type is used to represent a group of unordered statistical datapoints for calculations such as mean, median, and variance.

Args:

data (list): List or other iterable containing numeric values.

default (float): A value to be returned when a given statistical measure is not defined. 0.0 by default, but ``float('nan')`` is appropriate for stricter applications.

use_copy (bool): By default, Stats objects copy the initial data into a new list to avoid issues with modifications. Pass ``False`` to disable this behavior.

is_sorted (bool): Presorted data can skip an extra sorting step for a little speed boost. Defaults to False.

```
>>> from shared.statsutils import Stats

>>> myList = [1.234, 5.678, 9.012, 3.456, 7.890, 2.345, 4.567]

>>> s = Stats(myList)
```

clear_cache

"Stats" objects automatically cache intermediary calculations that can be reused. For instance, accessing the ``std_dev`` attribute after the ``variance`` attribute will be significantly faster for medium-to-large datasets. If you modify the object by adding additional data points, call this function to have the cached statistics recomputed.

count

The number of items in this Stats object. Returns the same as :func:`len` on a Stats object, but provided for pandas terminology parallelism.

```
>>> s.count

7
```

data

The data of the group in list form

describe(quantiles, format)

Provides standard summary statistics for the data in the Stats object, in one of several convenient formats.

Args:

quantiles (list): A list of numeric values to use as quantiles in the resulting summary. All values must be 0.0-1.0, with 0.5 representing the median. Defaults to ``[0.25, 0.5, 0.75]``, representing the standard quartiles.

format (str): Controls the return type of the function, with one of three valid values: ``"dict"`` gives back a :class:`dict` with the appropriate keys and values. ``"list"`` is a list of key-value pairs in an order suitable to pass to an OrderedDict or HTML table. ``"text"`` converts the values to text suitable for printing, as seen below.

Here is the information returned by a default ``describe``, as presented in the ``"text"`` format:

```
>>> stats = Stats(range(1, 8))
>>> print(stats.describe(format='text'))
count:    7
mean:     4.0
std_dev:  2.0
mad:      2.0
min:      1
0.25:     2.5
0.5:      4
0.75:     5.5
max:      7
```

get_quantile(quantile)

Get a quantile from the dataset. Quantiles are floating point values between ``0.0`` and ``1.0``, with ``0.0`` representing the minimum value in the dataset and ``1.0`` representing the maximum. ``0.5`` represents the median:

```
>>> Stats(range(100)).get_quantile(0.5)
49.5
```

get_zscore(value)

Get the z-score for *value* in the group. If the standard deviation is 0, 0 inf or -inf will be returned to indicate whether the value is equal to, greater than or below the group's mean.

```
>>> s.get_zscore(3)
-0.7145698838888683
```

histogram(bins, bin_digits, raw)

Produces a list of ``(bin, count)`` pairs comprising a histogram of the Stats object's data, using fixed-width bins. See: meth:`Stats.histogram` for more details.

Args:

bins (int): maximum number of bins, or list of floating-point bin boundaries. Defaults to the output of Freedman's algorithm.

bin_digits (int): Number of digits used to round down the bin boundaries. Defaults to 1.

raw (bool): return a single dimensioned list. Defaults to False

```
>>> s.histogram(bins = 6)
[(1.2, 2), (2.5, 1), (3.8, 1), (5.1, 1), (6.4, 0), (7.7, 2)]
>>> s.histogram(bins = 6, raw = 1)
[1.2, 1.2, 2.5, 3.8, 5.1, 7.7, 7.7]
```

iqr

Inter-quartile range (IQR) is the difference between the 75th percentile and 25th percentile. IQR is a robust measure of dispersion, like standard deviation, but safer to compare between datasets, as it is less influenced by outliers.

```
>>> s.iqr
3.8834999999999997
```

kurtosis

Indicates how much data is in the tails of the distribution. The result is always positive, with the normal "bell-curve" distribution having a kurtosis of 3.
http://en.wikipedia.org/wiki/Kurtosis

```
>>> s.kurtosis
2.058503273957092
```

mad, median_abs_dev

Median Absolute Deviation is a robust measure of statistical dispersion.
http://en.wikipedia.org/wiki/Median_absolute_deviation

```
>>> s.mad
2.222
```

max

The maximum value present in the data.

```
>>> s.max
9.012
```

mean

The arithmetic mean, or "average". Sum of the values divided by the number of values.

```
>>> s.mean
4.8831428571428575
```

median

The median is either the middle value or the average of the two middle values of a sample. Compared to the mean, it's generally more resilient to the presence of outliers in the sample.

```
>>> s.median
4.567
```

min

The minimum value present in the data.

```
>>> s.min
1.234
```

mode, all_modes

The mode is the value (or values) that occur the most number of times within the group. There is some dispute over whether an even distribution constitutes no mode, or they are all modes, so we have an attribute for each.

```
>>> s.mode
[]
>>> s.all_modes
[2.345, 3.456, 1.234, 9.012, 4.567, 7.89, 5.678]
```

pearson_type

Pearson distribution represents a system whereby for every member
the probability density function f(x) satisfies a differential equation
Returns the type of Pearson distribution.

https://www.vosesoftware.com/riskwiki/PearsonType5distribution.php

```
>>> s.pearson_type
1
```

rel_std_dev

Standard deviation divided by the absolute value of the average.
http://en.wikipedia.org/wiki/Relative_standard_deviation

```
>>> s.rel_std_dev
0.5396834857321365
```

skewness

Indicates the asymmetry of a curve. Positive values mean the bulk of the values are on the left side of the average and vice versa.

http://en.wikipedia.org/wiki/Skewness

```
>>> s.skewness
0.2749313814918259
```

std_dev

Standard deviation. Square root of the variance.

```
>>> s.std_dev
2.6353515584708416
```

trim_relative(amount)

A utility function used to cut a proportion of values off each end of a list of values. This has the effect of limiting the effect of outliers.

Args:

amount (float): A value between 0.0 and 0.5 to trim off of each side of the data. Default is 0.15

note:

This operation modifies the data in-place. It does not make or return a copy.

trimean

The trimean is a robust measure of central tendency, like the median, that takes the weighted average of the median and the upper and lower quartiles.

```
>>> s.trimean
4.704625
```

variance

Variance is the average of the squares of the difference between each value and the mean.

```
>>> s.variance
6.945077836734693
```